Claims 1, 2, 4, 32-36, 38, and 67-69 were once again rejected under 35 U.S.C. §102(b) as being anticipated by Hirsch. Applicants respectfully maintain their traversal of this rejection and further maintain their request that this rejection be reconsidered and withdrawn.

The claimed invention is directed towards a novel cryptographic key split combiner for assembling keys to enhance security against unauthorized data communications and further prevents reproduction of the key components thereof. Accordingly, the claimed cryptographic key split combiner includes a plurality of key split generators that generate cryptographic key splits and a key split randomizer that randomizes the cryptographic key splits to produce a cryptographic key. Each of the key split generators includes means for generating key splits from seed data.

In contrast, Hirsch discloses a software data protection mechanism, which includes an apparatus for generating a key. The description of the key generating apparatus is set forth in detail from column 3, line 47 through column 5, line 28, with reference to Figs. 1A-C. First, a 32-bit binary value is loaded into an input register 12, and each bit of the 32-bit value is provided as an input to a corresponding scrambler array container 140-n, n=0-31. Each container 140-n in the scrambler array 14 determines whether the input bit of that container is passed

to a respective bit position of a 32-bit output register 18, or whether the complement of that bit is passed instead.

The control for determining whether the input bit or its complement is passed is determined by every fifth bit of a pseudorandom sequence that is generated by a pseudorandom number generator 16. This generator takes a single seed and generates a single sequence that is loaded serially into the containers of the scrambler array 14, such that five bits are stored in each container. Because there are 32 containers, a 5-bit number also designates each bit position (00000, 00001, 00010, 00011, … , 11101, 11110, 11111). The result of an XOR operation on the LSB of the bit position and the LSB of the 5-bit pseudorandom sequence segment stored in the container determines whether the input bit or its complement will be provided to the output register. Because the LSB of the bit position alternates in the sequence of scrambler array containers, the XOR result for all even-numbered containers will be the pseudorandom number LSB, and the XOR result for all odd-numbered containers will be the pseudorandom number LSB complement. The actual bit position of the output register to which the respective "scrambled" input value bits is provided is determined by the actual value of the pseudorandom number, although the exact correspondence is not disclosed by Hirsch.

The 32-bit value stored in the output register 18 is now encoded by mapping the values of grouped bits according to a table. That is, the 32 bits of output data are separated into four groups of eight bits each. Each 8-bit group is stored in a set of two overlapping 5-bit registers 200-1a, 200-1b. The outputs of these registers (eight 5-bit values) are provided to an alphanumeric table 204, which maps the input values to provide eight alphanumeric values. The sequence consisting of these values is the key, which is stored in a user key register 24.

Thus, whereas Claim 1 recites a plurality of key split generators, each receiving seed data and generating a respective key split, and a randomizer for receiving and randomizing the plurality of key splits to generate a key, Hirsch discloses an array 14 for scrambling a single value according to a single pseudorandom sequence generated from a single seed, and an encoder 20 for mapping the scrambled value to generate the key. That is, contrary to the assertion set forth in the Examiner's response, Hirsch does not disclose a plurality of key split generators, each receiving seed data and generating a respective key split, as recited in Claim 1. Rather, Hirsch discloses a plurality of containers in a single scrambler array, which together receive a single serially-shifted pseudorandom sequence generated from a single seed. Further, Hirsch does not

disclose randomizing a plurality of separately generated key splits to generate a key, as recited in Claim 1. Rather, Hirsch discloses taking a single 32-bit scrambled value, and mapping 8-bit segments of that value according to a stored, predetermined table, to generate a key.

In summary, the combiner of Claim 1 takes separate key splits, generated based on separate seed values, and randomizes the splits to provide a key. In contrast, Hirsch takes a single input value, scrambles the value according to a single pseudorandom stream based on a single seed, and maps the scrambled value according to a fixed table to generate the key. Clearly, Hirsch describes an apparatus that is quite different from the combiner recited in Claim 1, and further does not even include any of the elements recited in Claim 1. Accordingly, Hirsch fails to anticipate, or even suggest, the presently claimed invention recited in Claim 1, as well as the invention of Claims 2, 4, and 32-34, all of which depend from Claim 1. The rejection of these claims, therefore, should be withdrawn.

Claim 35 recites a process for forming cryptographic keys. The process includes generating a plurality of cryptographic key splits from seed data; and randomizing the cryptographic key splits to produce a cryptographic key.

As set forth above, the claimed invention is not taught, or even suggested, by Hirsch. That is, Hirsch does

not generate a plurality of cryptographic key splits, but rather Hirsch generates a single scrambled value from a single 32-bit value. Furthermore, Hirsch does not generate key splits from seed data, but rather Hirsch merely generates a pseudorandom sequence from a seed, and the sequence is used to scramble the input value, not to generate a split on which the key is based as in the claimed invention. Further still, Hirsch does not disclose randomizing key splits to produce a cryptographic key, but rather Hirsch discloses mapping a single scrambled value to generate a key.

Thus, Hirsch does not teach, or even suggest, the process recited in independent Claim 35, or the invention recited in Claims 36 and 38, which depend from Claim 35. The rejection of these claims, therefore, should be withdrawn, as well.

Claim 66 recites a cryptographic key formed by the process of Claim 35. However, as set forth above, Hirsch discloses a different process, and a different apparatus for carrying out the key-generating process, and therefore fails to provide the presently claimed key. Hirsch, therefore, also fails to anticipate the invention recited in Claim 66, as well as the invention of Claims 67-69, which depend from Claim 66. The rejection of these claims, therefore, should also be withdrawn.

Therefore, since invalidity for anticipation requires that all of the elements and limitations of the rejected claims be found within a single prior art reference, and that there be no difference between the claimed invention and the reference disclosure (*Scripps Clinic & Research Foundation v. Genentech, Inc.* 18 USPQ2d 1001, 1010 (Fed. Cir. 1991)), it is respectfully submitted that a *prima facie* case of anticipation has not been presently established, and thus Applicants request that the present rejection under 35 U.S.C. §102(b) be withdrawn.

Claims 3 and 37 were rejected under 35 USC §103(a) as being unpatentable over Hirsch, in view of Albert et al. Applicants respectfully maintain their traversal to this rejection as well, and further maintain their request that this rejection also be reconsidered and withdrawn.

Claim 3 depends from Claim 1, and therefore also recites the combiner of Claim 1, wherein the plurality of key split generators includes a random split generator for generating a random split based on reference data, and the random split generator includes means for generating a random sequence based on the reference data. Therefore, the arguments set forth above distinguishing Claim 1 from the teachings of Hirsch are applied to the present rejection as well.

That is, Hirsch does not disclose the combiner recited in Claim 1. Furthermore, the teachings of Albert et al. are insufficient to compensate for the deficiencies of Hirsch, relative to the claimed invention. In particular, Albert et al. merely disclose a random number generator. There is no teaching or suggestion by Albert et al. that the disclosed random number generator should or could be used as part of a random split generator in a cryptographic key split combiner as recited in Claim 1. Further, Albert et al. does not overcome the deficiencies of Hirsch in disclosing the combiner recited in Claim 1. That is, Hirsch does not disclose a combiner that takes separate key splits that are generated based on separate seed values and randomizes the splits to provide a key, as recited in Claim 1. Albert et al., by merely describing a random number generator, fail to provide Hirsch with a means for generating separate key splits from seed data, or for randomizing separate splits to provide a key.

Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion supporting the combination. The mere fact that the prior art may be modified in the manner suggested by the Examiner does not make the modification obvious unless the prior art suggested the desirability of such modification (*In re Fritch*, 23

USPQ2d 1780, 1783-84 (Fed. Cir. 1992)). However, in the present rejection, the Office Action does not specify how the teachings of these two references could be combined, but rather merely implies that the random number generator described by Albert et al. could be substituted for the Hirsch pseudorandom number generator to provide less predictability. However, the Hirsch pseudorandom number generator does not provide a key split, that is, a component of the generated key. Rather, this generator provides one element used to scramble the single 32-bit input in the scrambler array that is the only component of the generated key. It is a functional input, rather than a key component.

Therefore, for at least the reasons set forth above, it is respectfully submitted that no combination of the teachings of the cited references could render obvious the invention recited in Claim 3, and therefore the rejection of Claim 3 should be withdrawn.

Claim 37 recites the process of Claim 35, wherein generating a plurality of cryptographic key splits includes generating a random key split based on reference data, and wherein generating a random key split includes generating a random sequence based on the reference data

As set forth above, Hirsch does not disclose the process recited in Claim 35. Such deficiencies of the primary reference are not compensated for by the teachings

of Albert et al. which merely disclose a random number generator. Thus, there is no suggestion by Albert et al. that the disclosed random number generator should or could be used to generate a random key split in the process recited in Claim 35. Further, Albert et al. does not overcome the deficiencies of Hirsch in disclosing the process recited in Claim 35. That is, Hirsch does not disclose generating a plurality of cryptographic key splits, generating key splits from seed data, or randomizing key splits to produce a cryptographic key, as recited in Claim 35. Hirsch generates a single scrambled value from a single 32-bit value, only generates a single pseudorandom sequence from a single seed, only uses the sequence to scramble the input value rather than to generate a split on which the key is based, and maps a single scrambled value to generate a key. Rather, Albert et al. merely describe a random number generator which fail to provide Hirsch with a means for both generating separate key splits from separate seed values and for randomizing separate splits to provide a key.

The present rejection does not specify how the teachings of these two references could be combined, but implied that the Albert et al. random number generator could be substituted for the Hirsch pseudorandom number generator to provide less predictability. However, the Hirsch pseudorandom number generator does not provide a key split,

that is, a component of the generated key. Rather, this generator provides one element used to scramble the single 32-bit input in the scrambler array that is the only component of the generated key. It is a functional input, rather than a key component.

Therefore, for at least the reason set forth above, it is respectfully submitted once again that the proposed combination of references fail to render obvious the invention recited in Claim 37. The rejection of Claim 37, therefore, should be withdrawn.

Claims 5 and 39 were rejected under 35 USC §103(a) as being unpatentable over Hirsch, in view of Thomlinson et al. Applicants respectfully maintain their traversal of this rejection, as well, and further maintain their request that this rejection also be reconsidered and withdrawn.

Claim 5 depends from Claim 1 and recites that the random split generator includes means for generating a key split based on the reference data and on chronological data.

Thomlinson et al. describe a non-biased pseudorandom number generator that includes an input device that gathers classes of bits provided elsewhere in a computer to hash a seed used to generate the number. One of the classes of bits is a machine class that relates to operating parameters of the computer, including time of day and date.

As set forth above, Hirsch does not disclose the combiner recited in Claim 1. Furthermore, such deficiencies, relative to the claimed invention, are not compensated for by Thomlinson et al. which merely describe a pseudorandom number generator. There is no suggestion by Thomlinson et al. that the disclosed number generator should or could be used as part of a random split generator in a cryptographic key split combiner as recited in Claim 1. Further, the Thomlinson et al. reference does not overcome the deficiencies of Hirsch in disclosing the combiner recited in Claim 1. That is, Hirsch does not disclose a combiner that takes separate key splits that are generated based on separate seed values, and randomizes the splits to provide a key, as recited in Claim 1. Thomlinson et al., by merely describing a pseudorandom number generator, do not provide Hirsch with a means for generating separate key splits from separate seed values, or for randomizing separate splits to provide a key, as in the claimed invention.

The present rejection does not specify how the teachings of these two references could be combined, but rather implies that the Thomlinson et al. pseudorandom number generator could be substituted for the Hirsch pseudorandom number generator, because the time and date component of the number would make the number harder to

guess. However, the Hirsch pseudorandom number generator does not provide a key split, that is, a component of the generated key. Rather, this generator provides one element used to scramble the single 32-bit input in the scrambler array that is the only component of the generated key. It is a functional input, rather than a key component. Thus, substituting the Thomlinson et al. pseudorandom number for the Hirsch pseudorandom number merely serves to replace one number that is not a key split for another number that is not a key split. As a result, the limitations of Claims 1 and 5 are still not satisfied.

Therefore, for at least the reasons set forth above, it is respectfully submitted that the proposed combination of references fail to render obvious the invention recited in Claim 5. The rejection of Claim 5, therefore, should be withdrawn.

Claim 39 depends from independent Claim 35, and recites that generating a random key split includes generating a key split based on the reference data and on chronological data.

As set forth above, Hirsch does not disclose the process recited in Claim 35. Furthermore, Thomlinson et al. merely disclose a pseudorandom number generator. There is no suggestion by Thomlinson et al. that the disclosed random number generator should or could be used to generate a random key split in the process recited in Claim 35.

Further still, the Thomlinson et al. reference does not overcome the deficiencies of Hirsch, in relation to the claimed invention, in disclosing the process recited in Claim 35. That is, Hirsch does not disclose generating a plurality of cryptographic key splits, generating key splits from seed data, or randomizing key splits to produce a cryptographic key, as recited in Claim 35. Hirsch generates a single scrambled value from a single 32-bit value, only generates a single pseudorandom sequence from a single seed, only uses the sequence to scramble the input value rather than to generate a split on which the key is based, and maps a single scrambled value to generate a key. Thomlinson et al. merely describe a pseudorandom number generator, thus failing to provide Hirsch with means for both generating separate key splits from separate seed values and randomizing separate splits to provide a key, as recited in Claim 35.

The present rejection does not specify how the teachings of these two references could be combined, but merely implies that the Thomlinson et al. pseudorandom number generator could be substituted for the Hirsch pseudorandom number generator, because the time and date component of the number would make the number harder to guess. However, the Hirsch pseudorandom number generator does not provide a key split, that is, a component of the

generated key. Rather, this generator provides one element used to scramble the single 32-bit input in the scrambler array that is the only component of the generated key. It is a functional input, rather than a key component. Thus, substituting the Thomlinson et al. pseudorandom number for the Hirsch pseudorandom number merely replaces one number that is not a key split for another number that is not a key split. As a result, the limitations of Claims 35 and 39 are still not satisfied.

Therefore, for at least the reasons set forth above, it respectfully submitted that the proposed combination of references fail to render obvious the invention recited in Claim 39. The rejection of Claim 39, therefore, should be withdrawn.

Claims 6, 7, 9-11, 13-16, 40, 41, 43-45, and 47-50 were rejected under 35 USC §103(a) as being unpatentable over Hirsch, in view of Ming et al. Applicants respectfully maintain their traversal of this rejection, and further maintain their request that this rejection be reconsidered and withdrawn.

Claims 6, 7, 9-11, and 13-16 all depend from Claim 1, and Claims 40, 41, 43-45, and 47-50 all depend from Claim 35, and therefore the arguments set forth above are applied to the present rejection as well.

As thoroughly discussed above, Hirsch does not disclose the invention recited in Claims 1 and 35. Ming et al. disclose encoder and decoder apparatus for a cable television signal having embedded viewer access control data. The Ming et al. invention provides means for blocking selected television programming classes in a cable television transmission, based on access privileges of a viewer at a television receiver having a decoder unit. Access denial codes are embedded in the transmitted video signal, and are read by the decoding unit, which implements the blocking function. Programs that the viewer is not authorized to access are scrambled, wherein the scrambling function takes the form of random line inversion. Thus, Ming et al. do not overcome the deficiencies of Hirsch, that is, Ming et al. do not even disclose a key split combiner. Thus, Applicants respectfully submit that the proposed combination of Hirsch and Ming et al. could render obvious the invention recited in Claims 6, 7, 9-11, 13-16, 40, 41, 43-45, and 47-50. However, the features recited in these dependent claims even further distinguish the claimed invention from the cited references.

Furthermore, on page 3 of the Office Action, it is asserted that the Ming et al. system would be inoperative if the seed values were not key splits. However, Applicants respectfully submit that this statement demonstrates a

fundamental misunderstanding of the claimed invention. That is, according to the claimed invention, the key splits are generated using the seed values. On the other hand, it appears that the Examiner misunderstands that the key splits are actually the seed values. Accordingly, it is respectfully submitted further that the teachings of Ming et al. have been inappropriately applied in rejecting the claimed invention, as is further discussed below.

With regards to Claims 6 and 40, the Examiner stated that Ming et al. disclose generating a key split based on static data, at column 4, lines 4-7. That passage describes generating and storing an initial seed value. This seed value is used in a pseudorandom sequence generator as the random function driving the line inverter. See column 14, lines 30- column 15, line 15. Thus, generating and storing the seed value by Ming et al. is not the same as generating a static key split. That is, Ming et al. have no key, but rather simply performs random line inversion. Even if the pseudorandom sequence were considered a key, it has no separate components, that is, no splits. Neither Hirsch nor Ming et al. discloses a randomizing key split combiner that generates a cryptographic key from a number of splits, which are generated from separate seeds. As a result, it is respectfully submitted that the proposed combination of references fail to render obvious the invention recited in

Claims 6 and 40. Accordingly, the rejection of these claims should be withdrawn.

Regarding Claims 7 and 41, the Examiner stated that Ming et al. disclose a means of updating the static data, at column 4, line 8. That passage describes generating a next seed value. This seed value is used to generate a new pseudorandom sequence for the subsequent video frame. As set forth above regarding the rejection of Claims 6 and 40, this is not an update of static data used in generating a key split. Therefore, the rejection of Claims 7 and 41 should be withdrawn.

Regarding Claims 9 and 43, the Examiner stated that Ming et al. disclose a token split generator for generating a token key split based on label data, at column 6, lines 26-29; column 5, lines 65-67; and column 6, lines 1-5. In those passages, Ming et al. disclose the use of five levels of viewer access. However, as set forth above, these are not key splits, and they have nothing to do with tokens. The access control data described by Ming et al. is embedded in the video data, and it is not used to generate a key split (see column 13, lines 16-20). Further, this data is not derived from a token. Therefore, the proposed combination of references fails to render obvious the invention recited in Claims 9 and 43, and thus the rejection of these claims should be withdrawn.

Regarding Claims 10 and 44, the Examiner stated that Ming et al. suggest reading label data from a storage medium, at column 7, lines 11-22. That passage describes that the decoder apparatus has a prestored user address that corresponds to a user address in the access control data. First, because the user address is stored at the decoder and compared with incoming address data, it could not be the basis for a key split that is generated and then randomized with other key splits. Further, the fact that it is prestored at the decoder apparatus demonstrates that this data could not be the basis for a token key split, even if it were part of some key split, and even if Ming et al. used any key splits at all. Therefore, the rejection of Claims 10 and 44 should also be withdrawn.

Regarding Claims 11 and 45, the Examiner stated that Ming et al. describe label data that includes user authorization data, at column 7, lines 22-25. That passage describes that the video data includes a program class code identifying authorized classes of users. However, this program class code is not the basis for key split data, but rather it is merely embedded in the video data, and is not a component of a cryptographic key. Further, even if it were a key split, there is no mention of a token, so it could not be the basis of a token key split. The rejection of Claims 11 and 45 should be withdrawn.

Regarding Claims 13 and 47, the Examiner stated that Ming et al. illustrate a means for generating a pseudorandom sequence, based on label data, at column 13, lines 45-50; Figure 2, items 113-115; and column 14, lines 39-44. In those passages, Ming et al. disclose two different processes. The column 13 and Figure 2 references describe how access control data from two different channel processors are alternately encrypted using the conventional DES algorithm. Thus, the data that the Examiner refers to as the label data is encrypted, but no pseudorandom sequence is generated. The column 14 passage describes the process taking place in Figure 3, which in part shows details of the data formatter and video scramble control block 118 in Figure 2. This block receives an 8-bit value 121 from an unknown source, and a line signal 122. Based on these signals, a pseudorandom sequence is generated by the generator 120. This sequence is used to drive the random line inverter. Thus, what the Examiner considers to be label data is encrypted, and data from a completely different source is used to generate a pseudorandom sequence. Therefore, the "label data" and the pseudorandom sequence are completely unrelated, and further are unrelated to the presently claimed key and key splits. The "label data" is encrypted for secure transmission to the decoder apparatus, and the sequence is used to directly scramble the

substantive data of the transmission in a manner that is unrelated to the access control data. Again, neither process involves any token, let alone a token split. Clearly, the requirements of Claims 13 and 47 are not satisfied, and therefore the rejection of these claims should be withdrawn.

Regarding Claims 14 and 48, the Examiner stated that Ming et al. specify the means for generating a key split based on label data and organization data, at column 6, lines 26-29 and 59-65; column 5, lines 65-67; and column 6, lines 1-5. In those passages, Ming et al. disclose the use of five levels of viewer access, based on different categories particular to users and classes of users. However, these are not key splits, and are unrelated to the claimed tokens. The access control data described by Ming et al. is embedded in the video data, and it is not used to generate a key split (see column 13, lines 16-20). This data, after being decrypted at the decoder, is the subject of a direct comparison to determine if access is granted. The data is not the basis for generating a split that will be randomized to form a key that will be used in the encryption process. Further, this data is not derived from a token. Thus, it is respectfully submitted that the proposed combination of references fail to render obvious

the invention recited in Claims 14 and 48, and therefore, the rejection of these claims should be withdrawn.

Regarding Claims 15 and 49, the Examiner stated that Ming et al. suggest means for generating a key split based on the label data and on static data, at column 4, lines 4-7. That passage describes generating and storing an initial seed value. This seed value is used in a pseudorandom sequence generator as the random function driving the line inverter (see column 14, lines 30- column 15, line 15). Thus, generating and storing the seed value by Ming et al. is not the same as generating a static key split, as is presently claimed. That is, Ming et al. have no key, but rather the reference merely performs random line inversion. In the alternative, even if the pseudorandom sequence were considered a key, it has no separate components, that is, no splits, as is presently claimed. Therefore, since neither Hirsch nor Ming et al. discloses a randomizing key split combiner that generates a cryptographic key from a number of splits, which are generated from separate seeds, it is respectfully submitted that the proposed combination of references fail to render obvious the invention recited in Claims 15 and 49. Therefore, the rejection of these claims should be withdrawn.

Regarding Claims 16 and 50, the present rejection states that Ming et al. disclose means for updating the

static data, at column 3, lines 65-67 and column 4, lines 1-4. This passage describes several differentiated layers used to determine access by a user to the cable television transmission, what the Examiner had previously referred to when discussing label data. However, this is not static data on which a key split is based as generated by a token key split generator. Further, it is not disclosed that this data is updated, as is presently claimed. As shown in Figure 2 of the reference, this data is encrypted and transmitted to the decoder apparatus. It is not used to generate a key split that is randomized with other key splits to generate a cryptographic key. Accordingly, the present rejection of Claims 16 and 50 should be withdrawn.

In summary, both the Hirsch invention and the Ming et al. invention are so different from the claimed invention that no combination of the teachings of these two references in an attempt to result in the claimed invention is suggested or possible. Furthermore, it is noted once again that obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion supporting the combination. The mere fact that the prior art may be modified in the manner suggested by the Examiner does not make the modification obvious unless the prior art suggested the desirability of such modification (*In re Fritch*, infra).

Therefore, it is respectfully submitted that a *prima facie* case of obviousness has not been established, and Applicants request that the present rejection be withdrawn.

Claims 8 and 42 were rejected under 35 U.S.C. §103(a) as being unpatentable over Hirsch, in view of Ming et al., and further in view of Anshel et al. Applicants respectfully maintain their traversal of this rejection, and further maintain the request that this rejection also be reconsidered and withdrawn.

The present rejection asserts that Ming et al., at column 4, lines 18-20, describe modifying the divisor of static data. However, close examination of this passage does not support this assertion. Rather, the passage sets forth a procedure for updating seed values for generating the pseudorandom sequence, based on an incremented frame count.

Anshel et al. disclose a cryptographic sequence generator. In the passage cited in the rejection, Anshel et al. describe use of randomly-selected prime numbers to generate a list of Jacobi symbols and, according to a public key, a subset of the symbols is chosen to encrypt a single input bit. The Jacobi symbols and the public key both feature terms having prime number divisors (see column 11, lines 8-53). In contrast, Claims 8 and 42 recite updating the static data by modifying a prime number divisor of the

static data. The static data is a basis for a random key split.

None of the cited references discloses random key splits. It follows that none of the cited references discloses updating a basis of a random key split, or performing the update by modifying a prime number divisor of the basis. Anshel et al. disclose the use of prime number divisors in a Jacobi sequence and in a public key. However, demonstrating that it is known to use prime number divisors in an application related to cryptography fails to suggest to one of ordinary skill in the art that modification of a prime number divisor of a basis of a random key split is beneficial in generating a key based on the randomization of a number of key splits, particularly in view of other references that do not even disclose the use of key splits. Thus, it is respectfully submitted that the proposed combination of references fail to render obvious the invention recited in Claims 8 and 42, and therefore it is submitted that the rejection of these claims should be withdrawn.

Applicants respectfully maintain their traversal to the rejection of Claims 12 and 46 under 35 U.S.C. §103(a) as being unpatentable over Hirsch, in view of Ming et al., and further in view of Albert et al., and further maintain their request that this rejection be reconsidered and withdrawn.

As set forth above, Hirsch does not disclose the invention recited in Claims 1 and 25. The present rejection states that Ming et al. disclose generating a pseudorandom sequence based on label data. However, as set forth above regarding the rejection of Claims 13 and 47, this characterization of the reference is not correct. The access control data is encrypted for transmission to the decoder apparatus, and different data is used to generate a pseudorandom sequence to drive a line inverter. Further, Albert et al. merely disclose a design for a random number generator. Albert et al. do not disclose or suggest anything that would lead one of ordinary skill in the art, in view of the other references, to develop a cryptographic key split combiner as recited in Claims 12 and 46. Therefore Applicants respectfully submit that the rejection of these claims should be withdrawn.

Claims 17 and 51 were rejected under 35 U.S.C. §103(a) as being unpatentable over Hirsch, in view of Ming et al., and further in view of Anshel et al. Applicants respectfully maintain their traversal of this rejection, and further maintain their request that this rejection be reconsidered and withdrawn.

The present rejection states that Ming et al. (column 4, lines 18-20) describe modifying the divisor of static data. However, this characterization of the reference is

not correct. Rather, the passage sets forth a procedure for updating seed values for generating the pseudorandom sequence, based on an incremented frame count.

Anshel et al. disclose a cryptographic sequence generator. In the passage cited in the rejection, Anshel et al. describe use of randomly-selected prime numbers to generate a list of Jacobi symbols and, according to a public key, a subset of the symbols is chosen to encrypt a single input bit. The Jacobi symbols and the public key both feature terms having prime number divisors (see column 11, lines 8-53). On the other hand, Claims 17 and 51 recite updating the static data by modifying a prime number divisor of the static data. The static data is a basis for a random key split.

Furthermore, none of the presently cited references discloses random key splits. As a result, none of the cited references discloses updating a basis of a random key split, or performing the update by modifying a prime number divisor of the basis. Anshel et al. disclose the use of prime number divisors in a Jacobi sequence and in a public key. However, demonstrating that it is known to use prime number divisors in an application related to cryptography is not enough to suggest to one of ordinary skill in the art that modification of a prime number divisor of a basis of a random key split is beneficial in generating a key based on

the randomization of a number of key splits, particularly in view of other references that do not even disclose the use of key splits.

Accordingly, Applicants submit that the proposed combination of references fails to render obvious the invention recited in Claims 17 and 51, and therefore the rejection of these claims should be withdrawn.

Applicants respectfully maintain their traversal of the rejection of Claims 18, 20-24, 52, and 54-58 under 35 U.S.C. §103(a) as being unpatentable over Hirsch, in view of Ming et al., and further in view of Anshel et al., and further maintain their request that this rejection be reconsidered and withdrawn.

Regarding Claims 18 and 52, the present rejection states that Anshel et al. discuss a console split generator for generating a console key split based on maintenance data, citing column 8, lines 8-15. However, the cited passage actually describes a conventional public key infrastructure arrangement. As disclosed, users of a network are each given a fixed (line 10) private key, and a public key is broadcast to users of the network. In the reference, key splits of any kind are not disclosed. As disclosed, the private key is fixed, and such disclosure would actually teach away from having a key component that would depend on maintenance data, as in the claimed

invention. According to Anshel et al., this private key remains fixed, while each public key is used only once and then discarded, and a new public key is generated, based solely on an incremented state value. There is no seed provided to generate a split from maintenance data, or randomizer for combining a console split with other splits to determine a key, as recited in Claims 18 and 52. Rather, the key is determined directly from the state data.

As previously discussed, Hirsch and Ming et al. do not disclose the elements of Claims 1 and 35, from which Claims 18 and 52 respectively depend. It is acknowledged that these references also do not disclose the particular features of Claims 18 and 52. Thus, the proposed combination of references fail to render obvious the invention recited in Claims 18 and 52, and therefore the present rejection should be withdrawn.

Regarding Claims 20 and 54, the present rejection asserts that Anshel et al. disclose means for generating a pseudorandom sequence based on maintenance data (column 8, lines 8-15). However, as described in the passage referring to Figure 4, the ZPNG generates a pseudorandom code which is transformed by a zeta code transformer to produce the public key. However, this description is unrelated to a split generator that generates one of a number of key splits that are randomized to form a cryptographic key, as in the

claimed invention. Rather, according to the reference, the sequence itself is transformed to become the key.

As set forth above, Hirsch and Ming et al. do not disclose such a combiner, so there is no motivation to apply the teachings of Anshel et al. to these two references to show that the Anshel et al. key could be a key split in a Hirsch/Ming et al. system. Thus, the proposed combination of references fail to render obvious the invention recited in Claims 20 and 54, and therefore, the present rejection should be withdrawn.

Regarding Claims 21 and 55, the present rejection states that Anshel et al. specify generating a key split based on previous and current maintenance data, citing column 8, lines 26 and 27. As described in the passage, a public key is generated based on a current state value. The key is used once and then discarded. The state value is incremented, and a new public key is determined based on this incremented value. Thus, the public key is determined based only on a current state value. However, the previous state value and current state value are never both used to determine the public key, as recited in Claims 21 and 55. See also Figure 4. Further, the key described by Anshel et al. is not a key split, as discussed previously. Further still, as set forth above, Hirsch and Ming et al. do not disclose the claimed combiner, and it is acknowledged in the

Office Action that these references do not disclose the particular features recited in Claims 21 and 55. Thus, the proposed combination of references fail to render obvious the invention recited in Claims 21 and 55, and therefore the present rejection should be withdrawn.

Regarding Claims 22 and 56, the present rejection states that Anshel et al. describe generating a key split based on maintenance data and static data, citing column 8, lines 16-22. However, the cited passage actually describes generation of the public key of a conventional public key infrastructure arrangement, and the public key is generated based on an incremented state value. Notably, there is no seed provided to generate a split from maintenance data, or randomizer for combining a console split with other splits to determine a key, as recited in Claims 22 and 56. Rather, the key is determined directly from the state data.

As set forth above, Hirsch and Ming et al. do not disclose the elements of Claims 1 and 35, from which Claims 22 and 56 respectively depend. That is, like Anshel et al., Hirsch and Ming et al. do not disclose a number of key splits, generated from different seeds, that are randomized to generate a key. It is acknowledged in the Office Action that these references also do not disclose the particular features of Claims 22 and 56. Thus, the proposed combination of references fail to render obvious the

invention recited in Claims 22 and 56, and therefore the present rejection should be withdrawn.

Regarding Claims 23 and 57, the present rejection states that Anshel et al. delineate a means for updating static data, citing column 8, lines 8, 26, and 27. However, this passage merely describes generating a new public key based on an incremented state value. The disclosed state data has been previously identified with the claimed maintenance data. Claims 23 and 57, through their respective dependence from Claims 22 and 56, require both maintenance data and static data. These individual elements are not delineated in the Anshel et al. invention. That is, a close reading of the reference, and of the cited passage in particular, does not reveal a correspondence of disclosed elements with the presently claimed elements. The lack of detail in the assertions in the present rejection do not assist in making this identification. In any case, Anshel et al. does not disclose both maintenance and static data, which is updated, as components of a key split that is randomized with other key splits to produce a key, as recited in the claims. As set forth above, the other cited references also fail to disclose these elements of the claimed invention. Thus, the proposed combination of references fail to render obvious the invention recited in

Claims 23 and 57, and therefore the present rejection should be withdrawn.

Regarding Claims 24 and 58, the present rejection states that Anshel et al. illustrate that updating the static data includes modifying a prime number divisor of the static data, citing column 11, lines 8-25 and Figure 8, item 71. In the cited passage, Anshel et al. describe use of randomly-selected prime numbers to generate a list of Jacobi symbols and, according to a public key, a subset of the symbols is chosen to encrypt a single input bit. The Jacobi symbols and the public key both feature terms having prime number divisors (see column 11, lines 8-53). However, the described process in not relevant to anything that the Examiner has identified as being static data, or to the updating of this static data.

On the other hand, Claims 24 and 58 recite updating the static data by modifying a prime number divisor of the static data. Demonstrating that it is known to use prime number divisors in an application related to cryptography is not enough to suggest to one of ordinary skill in the art that modification of a prime number divisor of static data basis of a key split is beneficial in generating a key based on the randomization of a number of key splits, particularly in view of other references that do not even disclose the use of key splits. Thus, the proposed combination of

references fail to render obvious the invention recited in Claims 24 and 58, and therefore the present rejection should be withdrawn.

Applicants respectfully maintain their traversal of the rejection of Claims 19 and 53 under 35 U.S.C. §103(a) as being unpatentable over Hirsch, in view of Anshel et al., and further in view of Albert et al., and further maintain their request that this rejection be reconsidered and withdrawn.

The present rejection asserts that Anshel et al. describe means for generating a pseudorandom sequence based on maintenance data (column 8, lines 8-15), and that Albert et al. specify a random sequence (column 1, line 66 - column 2, line 2).

First, it is noted that Claims 19 and 53 require generating a console key split, among other key splits, that are to be randomized to produce a key. The console key split is based on maintenance data, and the console split generator (or generating the console split) includes generation of a random sequence based on the maintenance data. As set forth above, Hirsch does not disclose any of these elements. Anshel et al. describe generating a public key from a pseudorandom sequence based on a state value. However, Anshel et al. and Albert et al. are insufficient to compensate for such deficiencies since Anshel et al. do not

disclose a key generated by randomizing a number of key splits, and Albert et al. merely describe a circuit for generating a random sequence. Thus, none of the references discloses or suggests all the elements of Claims 19 and 53, including generating a console key split, among other key splits, that are to be randomized to produce a key, where the console key split is based on maintenance data, and the console split generator (or generating the console split) includes generation of a random sequence based on the maintenance data.

Therefore, the proposed combination of the teachings of the cited references fail to render obvious the invention recited in Claims 19 and 53, and therefore the present rejection should be withdrawn.

Applicants respectfully maintain their traversal of the rejection of Claims 25, 27-31, 59, and 61-65 under 35 U.S.C. §103(a) as being unpatentable over Hirsch, in view of Tomko et al., and further maintain their request that this rejection be reconsidered and withdrawn.

Regarding Claims 25 and 59, the present rejection states that Tomko et al. elaborate on a biometric split generator for generating a biometric key split based on biometric data. Tomko et al. disclose a fingerprint controlled public key cryptographic system. Tomko et al. utilize data derived from a user's fingerprint to generate a

private key to be used in encrypting and decrypting messages. The fingerprint data is provided as an input to a pseudorandom sequence generator to generate the key. However, no other components are included in the key, and therefore the biometric data used in the Tomko et al. system is not used to generate a biometric key split, as recited in the claims, but rather is used to generate the key itself. No other component is randomized with the biometric data to derive the key. As set forth above, Hirsch does not disclose a randomizer for combining key splits derived from individual seeds to generate a key, as recited in Claims 1 and 35. Thus, the proposed combination of references fails to render obvious the invention recited in Claims 25 and 59, and therefore the present rejection should be withdrawn.

Regarding Claims 27 and 61, the present rejection states that Tomko et al. disclose means for generating a pseudorandom sequence based on the biometric data. However, Claims 27 and 61 require biometric split generation, which, as noted above in discussing the rejection of Claims 25 and 59, is not disclosed by Tomko et al. As set forth above, Hirsch does not disclose a randomizer for combining key splits derived from individual seeds to generate a key, as recited in Claims 1 and 35, from which Claims 27 and 61 depend. Thus, the proposed combination of references fails

to render obvious the invention recited in Claims 27 and 61, and therefore the present rejection should be withdrawn.

Regarding Claims 28 and 62, the present rejection assert that Tomko et al. delineate means for generating a key split based on biometric data vectors and on biometric combiner data. However, Tomko et al. do not disclose generating a key split at all. Rather, Tomko et al. disclose generating a key based solely on biometric data, not on randomized splits. As set forth above, Hirsch does not disclose a randomizer for combining key splits derived from individual seeds to generate a key, as recited in Claims 1 and 35, from which Claims 28 and 62 depend. Thus, the proposed combination of references fails to render obvious the invention recited in Claims 28 and 62, and therefore the present rejection should be withdrawn.

Regarding Claims 29 and 63, the present rejection states that Tomko et al. explain a means for generating a key split based on biometric data and on static data. However, Tomko et al. do not disclose generating a key split at all. Rather, Tomko et al. disclose generating a key based solely on biometric data, not on randomized splits. As set forth above, Hirsch does not disclose a randomizer for combining key splits derived from individual seeds to generate a key, as recited in Claims 1 and 35, from which Claims 29 and 63 depend. Thus, the proposed combination of

references fails to render obvious the invention recited in Claims 29 and 63, and therefore the present rejection should be withdrawn.

Regarding Claims 30 and 64, the present rejection states that Tomko et al. illustrate updating the static data. However, Tomko et al. do not disclose generating a biometric key split, or any key split at all, as recited in the claims. Rather, Tomko et al. disclose generating a key based solely on biometric data, not on randomized splits. As set forth above, Hirsch does not disclose a randomizer for combining key splits derived from individual seeds to generate a key, as recited in Claims 1 and 35, from which Claims 30 and 64 depend. Thus, the proposed combination of references fails to render obvious the invention recited in Claims 30 and 64, and therefore the present rejection should be withdrawn.

Regarding Claims 31 and 65, the present rejection states that Tomko et al. elaborate that the means for updating the static data includes means for modifying the prime number divisor of the static data (column 7, line 45 - column 8, line 12). However, this passage describes the procedure used to derive an array *b* that is related to the unique number *u*, which is derived from the Fourier transform of the user's fingerprint data. That is, the passage describes the modular mathematics used to derive the

coefficients *b* that determine *u*. Prime numbers are not mentioned at all, and certainly not in terms of modifying the prime number divisor of any static data. No primes are used, and the only data used is derived from the biometric data.

Further, Tomko et al. do not disclose generating a key split at all. Rather, Tomko et al. disclose generating a key based solely on biometric data, not on randomized splits. As set forth above, Hirsch does not disclose a randomizer for combining key splits derived from individual seeds to generate a key, as recited in Claims 1 and 35, from which Claims 31 and 65 depend. Thus, the proposed combination of references fails to render obvious the invention recited in Claims 31 and 65, and therefore the present rejection should be withdrawn.

Applicants respectfully maintain their traversal of the rejection of Claims 26 and 60 under 35 U.S.C. §103(a) as being unpatentable over Hirsch, in view of Tomko et al., and further in view of Albert et al., and further maintain their request that this rejection be reconsidered and withdrawn.

Claims 26 and 60 recite generating (and means for generating) a random sequence based on biometric data for biometric split generation. It is asserted in the present rejection that Tomko et al. discuss means for generating a pseudorandom sequence based on biometric data, and that

Albert et al. specify a random sequence. However, Tomko et al. do not disclose generating a key split. Rather, Tomko et al. disclose generating a key based solely on biometric data, not on randomized splits. Furthermore, Albert et al. merely describe a random sequence generator. As discussed above, Hirsch does not disclose a randomizer for combining key splits derived from individual seeds to generate a key, as recited in Claims 1 and 35, from which Claims 26 and 60 depend. Thus, the proposed combination of references fails to render obvious the invention recited in Claims 26 and 60, and therefore the present rejection should be withdrawn.

All objections and rejections having been addressed, it is respectfully submitted that the present application is now in condition for allowance, and a Notice to that effect is earnestly solicited.

Respectfully submitted,

June 5, 2000

RABIN & CHAMPAGNE, P.C

David S. Lee
Registration No. 38,222

RABIN & CHAMPAGNE, P.C.
1725 K Street, N.W.
Suite 1111
Washington, D.C. 20006
Telephone : (202) 659-1915
Telefax : (202) 659-1898

DSL/dbp